



US006763384B1

(12) **United States Patent**
Gupta et al.

(10) Patent No.: **US 6,763,384 B1**
(45) Date of Patent: **Jul. 13, 2004**

(54) **EVENT-TRIGGERED NOTIFICATION OVER A NETWORK**

(75) Inventors: **Manish Gupta**, New Delhi (IN);
Natwar Modani, New Delhi (IN);
Parul A Mittal, New Delhi (IN)

(73) Assignee: **International Business Machines Corporation**, Armonk, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 750 days.

(21) Appl. No.: **09/613,524**

(22) Filed: **Jul. 10, 2000**

(51) Int. Cl.⁷ **G06F 15/173**

(52) U.S. Cl. **709/224; 709/226**

(58) Field of Search **709/224, 205, 709/223, 203, 235, 226, 318; 705/14, 1**

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,050,104 A * 9/1991 Heyen et al. 345/556
6,185,613 B1 * 2/2001 Lawson et al. 709/224
6,480,883 B1 * 11/2002 Tsutsumitake 709/203
6,591,279 B1 * 7/2003 Emens et al. 707/201

* cited by examiner

Primary Examiner—Ario Etienne

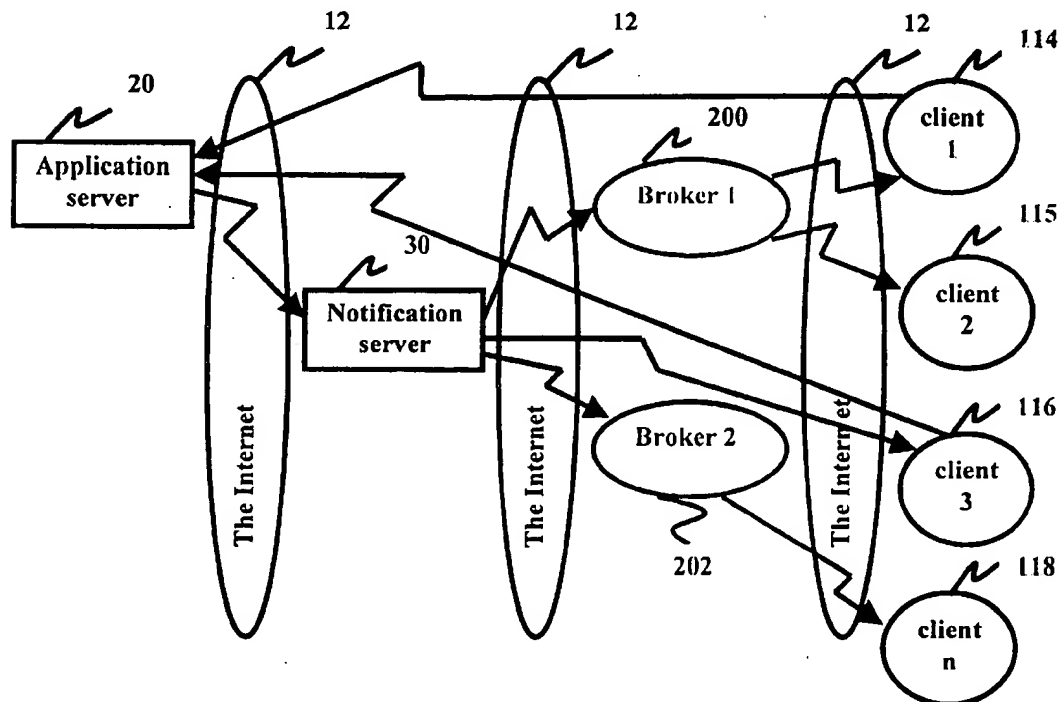
Assistant Examiner—Sahera Halim

(74) *Attorney, Agent, or Firm*—McGinn & Gibb, PLLC; T. Rao Coca, Esq.

(57) **ABSTRACT**

Internet services that rely on the transfer of real-time information have proliferated in recent years. Given the amount of information generated on the Internet, a user may request to be notified whenever particular information becomes available, such as the current highest bid in an on-line auction. The user enrolls to receive one or more messages and sends a list of desired messages through a client process running on the user's computer system. When the user is ready to receive messages, the client process connects to a notification server and provides its receiving address identifier. It then disconnects from the notification server and waits for messages. A message monitor detects the occurrence of events that trigger notifications, and informs the notification server of the event. The notification server determines the appropriate recipient for the notification, and sends it using a server-initiated end-to-end message transfer mechanism to the clients' receiving address identifiers. The notification can also be used as a control signal to control one or more applications, or to control an appliance remotely over a network. The notification server may make use of load-sharing devices or proxy servers to distribute the notification to the client processes.

29 Claims, 6 Drawing Sheets



US-PAT-NO: 6763384

DOCUMENT-IDENTIFIER: US 6763384 B1

TITLE: Event-triggered notification over a network

Brief Summary Text - BSTX (9):

Another notification mechanism is "client polling" wherein the client periodically polls a server for new information. Again, the information is being pulled by the client, but this may be done by an application running automatically on the user's system, without any intervention by the user. An example of this mechanism can be found in a website which uses an HTTP refresh mechanism to automatically refresh a web page displaying the latest scores for cricket matches every ninety seconds. Such polling schemes overcome the problem of user intervention by pulling information automatically on a periodic basis. However, these schemes have several disadvantages. The most important is bandwidth wastage. If the polling interval is kept small, the information content of the page is unlikely to change within one polling interval, and hence bandwidth resources are wasted in sending the fruitless client polling request. On the other hand, if the polling interval is long, the information may not get delivered to the user in time. If too many clients are polling a server frequently, there can be an enormous load on that server. In consequence, these polling schemes may not be highly scalable.



US006763384B1

(12) **United States Patent**
Gupta et al.

(10) Patent No.: **US 6,763,384 B1**
(45) Date of Patent: **Jul. 13, 2004**

(54) **EVENT-TRIGGERED NOTIFICATION OVER A NETWORK**

(75) Inventors: **Manish Gupta, New Delhi (IN);
Natwar Modani, New Delhi (IN);
Parul A Mittal, New Delhi (IN)**

(73) Assignee: **International Business Machines Corporation, Armonk, NY (US)**

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 750 days.

(21) Appl. No.: **09/613,524**

(22) Filed: **Jul. 10, 2000**

(51) Int. Cl.⁷ **G06F 15/173**

(52) U.S. Cl. **709/224; 709/226**

(58) Field of Search **709/224, 205,
709/223, 203, 235, 226, 318; 705/14, 1**

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,050,104 A * 9/1991 Heyen et al. 345/556
6,185,613 B1 * 2/2001 Lawson et al. 709/224
6,480,883 B1 * 11/2002 Tsutsumitake 709/203
6,591,279 B1 * 7/2003 Emens et al. 707/201

* cited by examiner

Primary Examiner—Ario Etienne

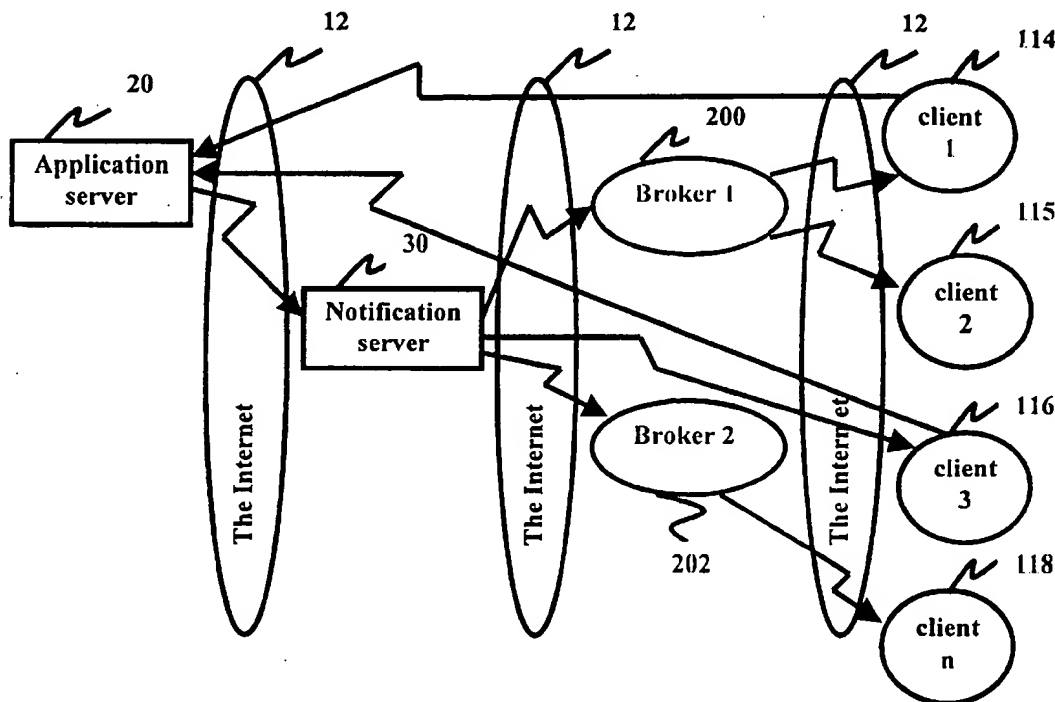
Assistant Examiner—Sahera Halim

(74) Attorney, Agent, or Firm—**McGinn & Gibb, PLLC; T. Rao Coca, Esq.**

(57) **ABSTRACT**

Internet services that rely on the transfer of real-time information have proliferated in recent years. Given the amount of information generated on the Internet, a user may request to be notified whenever particular information becomes available, such as the current highest bid in an on-line auction. The user enrolls to receive one or more messages and sends a list of desired messages through a client process running on the user's computer system. When the user is ready to receive messages, the client process connects to a notification server and provides its receiving address identifier. It then disconnects from the notification server and waits for messages. A message monitor detects the occurrence of events that trigger notifications, and informs the notification server of the event. The notification server determines the appropriate recipient for the notification, and sends it using a server-initiated end-to-end message transfer mechanism to the clients' receiving address identifiers. The notification can also be used as a control signal to control one or more applications, or to control an appliance remotely over a network. The notification server may make use of load-sharing devices or proxy servers to distribute the notification to the client processes.

29 Claims, 6 Drawing Sheets



US-PAT-NO: 6763384
DOCUMENT-IDENTIFIER: US 6763384 B1
TITLE: Event-triggered notification over a network

Abstract Text - ABTX (1):

Internet services that rely on the transfer of real-time information have proliferated in recent years. Given the amount of information generated on the Internet, a user may request to be notified whenever particular information becomes available, such as the current highest bid in an on-line auction. The user enrolls to receive one or more messages and sends a list of desired messages through a client process running on the user's computer system. When the user is ready to receive messages, the client process connects to a notification server and provides its receiving address identifier. It then disconnects from the notification server and waits for messages. A message monitor detects the occurrence of events that trigger notifications, and informs the notification server of the event. The notification server determines the appropriate recipient for the notification, and sends it using a server-initiated end-to-end message transfer mechanism to the clients' receiving address identifiers. The notification can also be used as a control signal to control one or more applications, or to control an appliance remotely over a network. The notification server may make use of load-sharing devices or proxy servers to distribute the notification to the client processes.

Brief Summary Text - BSTX (6):

One area which has seen rapid growth is that of on-line auctions. Goods are offered for auction at many sites on the World Wide Web. If an Internet user desires an article currently being auctioned, he or she may place a bid on-line. As the auction of the article in question may extend for several days, there is a need to track the current highest bid over an extended period. The bidder may wish to be notified whenever the highest bid changes.

Brief Summary Text - BSTX (8):

The most elementary notification mechanism is based on the "pull" technology where a client explicitly pulls new information from a server. Examples of websites using this technology for on-line auctions are known. Such systems tell the user to click on a reload button to see the latest information, for example the current highest bid in open-outcry auctions. This method has several disadvantages. It requires explicit user intervention to obtain the latest information, and this can be quite time consuming for the user. Also, since the user cannot correctly guess when the information content will change, this method leads to a waste of bandwidth resources when the user sends a request but the information has not changed. On the other hand, if the user waits too long before requesting the information it might be received too late; for example, after an auction has ended.

Brief Summary Text - BSTX (9):

Another notification mechanism is "client polling" wherein the client periodically polls a server for new information. Again, the information is being pulled by the client, but this may be done by an application running automatically on the user's system, without any intervention by the user. An example of this mechanism can be found in a website which uses an HTTP refresh mechanism to automatically refresh a web page displaying the latest scores for cricket matches every ninety seconds. Such polling schemes overcome the problem of user intervention by pulling information automatically on a periodic basis. However, these schemes have several disadvantages. The most important is bandwidth wastage. If the polling interval is kept small, the information content of the page is unlikely to change within one polling interval, and hence bandwidth resources are wasted in sending the fruitless client polling request. On the other hand, if the polling interval is long, the information may not get delivered to the user in time. If too many clients are polling a server frequently, there can be an enormous load on that server. In consequence, these polling schemes may not be highly scalable.

Brief Summary Text - BSTX (11):

Other system use email as a notification mechanism. There are several well-known examples of websites which allow users to receive email alerts for various predefined events. These email notifications do not have bandwidth wastage problems, but they cannot be regarded as real-time in nature, as there is no guaranteed upper limit to the delay between transmission and reception. The final portion of the path from the destination mail server to the client application uses either a polling scheme, or maintains an open connection. Furthermore, the highest bid update in an open-outcry au**ction** should not be sent by email, as this would lead to a very large number of emails being sent to the user, wasting time and causing much frustration. This kind of frequently changing data is not suited to the email world.

Detailed Description Text - DETX (22):

An end user is browsing a website that publishes a list of messages or events that the user can choose to be notified of in real-time. This could be, for example, an on-line open-outcry au**ction** site that can notify users whenever there is a change in the current highest bid. The user decides to enrol to receive one or more messages and sends a list of desired messages to the website through a client process 110 running on the user's computer system 50. The client process 110 may disconnect after sending the list of messages if the user does not want to start receiving messages immediately. Once the user wants to start receiving messages, the client process 110 connects to the notification server 30 and sends a message which comprises the identity of the client process 110 together with its receiving address identifier. The notification server 30 updates the databank of clients 110-118 that are currently on-line. The client process 110 then disconnects from the notification server 30 and waits for messages on the receiving address identifier.

Detailed Description Text - DETX (25):

In order to reduce the amount of data that needs to be sent with each notification, the transmitted message need contain only the changed data, for example, the amount of the winning bid for an au**ction** site. The client then dynamically generates a display incorporating the changed data for the user to view. An example of this is when data is sent in XML (eXtensible Markup Language). XML data contains only information regarding the content and structure of a message. It contains no information relating to the visual presentation of the message. In order to display the information to the user in a web browser 100, the XML data needs to be transformed into HTML (HyperText Markup Language) using technology such as CSS (Cascading Style Sheets) or XSL (eXtensible Stylesheet Language). The conversion from XML to HTML can be performed by the client 114-118, thus shifting some processing load from the application servers 20-24 to the clients 114-118. Since this reduces the amount of data that needs to be sent in every notification, the overall scalability of the system is enhanced.

Detailed Description Paragraph Table - DETL (1):

TABLE 1 Message Databank Client ID Message of Interest Bill Start of an Au**ction** on an "IBM Thinkpad" Bill Close of the Au**ction** on a "PowerPC" Fred IBM Stock Price change of more than 5% Joe Change in the highest bid in the "IBM Thinkpad" au**ction**



US006591266B1

(12) **United States Patent**
Li et al.

(10) **Patent No.: US 6,591,266 B1**
(45) **Date of Patent: Jul. 8, 2003**

(54) **SYSTEM AND METHOD FOR INTELLIGENT CACHING AND REFRESH OF DYNAMICALLY GENERATED AND STATIC WEB CONTENT**

(75) **Inventors:** Wen Syan Li, Fremont, CA (US);
Kasim Selouk Candan, Mountain View, CA (US); Divyakant Agrawal, Santa Barbara, CA (US)

(73) **Assignee:** NEC Corporation, Tokyo (JP)

(*) **Notice:** Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 395 days.

(21) **Appl. No.:** 09/639,208

(22) **Filed:** Aug. 14, 2000

Related U.S. Application Data

(60) Provisional application No. 60/218,418, filed on Jul. 14, 2000.

(51) **Int. Cl.⁷** G06F 17/30; G06F 7/00

(52) **U.S. Cl.** 707/10; 707/2; 707/9; 707/104.1

(58) **Field of Search** 707/4-10, 100-104.1

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,941,957 A * 8/1999 Ingrassia et al. 709/248

5,951,643 A * 9/1999 Shelton et al. 709/227
5,951,652 A * 9/1999 Ingrassia et al. 709/248
5,954,798 A * 9/1999 Shelton et al. 709/224
6,035,332 A * 3/2000 Ingrassia et al. 709/224
6,490,575 B1 * 12/2002 Berstis 707/3

* cited by examiner

Primary Examiner—Greta Robinson

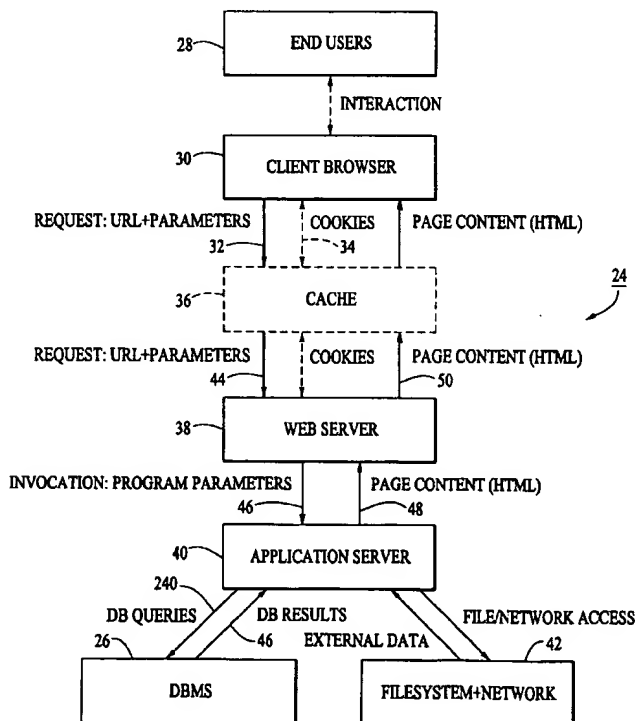
Assistant Examiner—Linh Black

(74) *Attorney, Agent, or Firm*—Foley & Lardner

(57) **ABSTRACT**

A system for updating Web pages stored in cache based on modifications to data stored in a database is disclosed. The system for updating stored Web pages may be part of a larger system having a database management system for storing data used to generate Web pages. The database management system is capable of identifying modified data stored in the database. The system for updating stored Web pages is comprised of one or more servers programmed for maintaining associations between the stored Web pages and the stored data, and receiving the identity of modified data from the memory management system. In addition, the servers are capable of determining, from the identified modified data and the maintained associations, which stored Web pages are associated with the identified modified data. Furthermore, the servers are capable of communicating an update command to the cache that contains the stored Web pages associated with the identified modified data, for the purpose of updating the stored Web pages.

164 Claims, 20 Drawing Sheets



US-PAT-NO: 6591266
DOCUMENT-IDENTIFIER: US 6591266 B1
TITLE: System and method for intelligent caching and refresh of dynamically generated and static web content

TITLE - TI (1):

System and method for intelligent caching and refresh of dynamically generated and static web content

Parent Case Text - PCTX (2):

Embodiments of the present invention claim priority from a U.S. Provisional Application entitled "Method and Apparatus for Intelligent Caching and Refresh of Dynamically Generated and Static Web Contents," Serial No. 60/218,418, filed Jul. 14, 2000, the contents of which are incorporated by reference herein.

Brief Summary Text - BSTX (8):

Many approaches and architectures have been proposed for CDS and most of these services, systems, and architectures have focused on static content. For example, CachePortal.TM. is a system providing CDS, and is described in pending U.S. patent application Ser. No. 09/545,805, entitled "System and Method for Efficient Content Delivery," filed Apr. 7, 2000, the contents of which are incorporated herein by reference. CachePortal.TM. has access to mirror servers which are used as edge cache to provide content to end users with less network delay. CachePortal.TM. can distribute content among mirror servers as well as remove, refresh, or invalidate the content in the mirror servers. CachePortal.TM. can also modify the attributes of content in the mirror servers. For example, CachePortal.TM. may check whether an object has been updated. If CachePortal.TM. finds that it is not updated, then CachePortal.TM. may change the value of the refresh time stamp or last modified date time stamp.

Brief Summary Text - BSTX (19):

One way to increase the probability that the Web pages in the cache 36 are fresh is to periodically refresh the pages through the Web server 38. However, this may result in many unnecessary requests to the Web server 38, the application server 40, and the DBMSs 26, and even with periodic refreshing, the Web pages in cache 36 cannot be guaranteed to be fresh. Because of the difficulty in ensuring that the data stored in cache 36 is fresh, important dynamic Web pages that are stored in cache 36 are typically set to expire immediately. Alternatively, such Web pages may be designated as non-cacheable.

Detailed Description Text - DETX (5):

Embodiments of the present invention include methods and systems for intelligently refreshing dynamically generated Web content in the cache as well as in the Web server. These methods and system architectures can also be applied to static content refresh to ensure all contents in the cache are consistent with the contents in the applications and the DBMSs.

Detailed Description Text - DETX (13):

CachePortal.TM. 94 will then use the URL/relevant operation mapping table 96 to determine which URLs are associated with the potentially affected queries 102. The Web pages associated with these URLs will then be have to be invalidated or refreshed. As a provider of content delivery services, CachePortal.TM. 94 keeps track of where Web pages have been stored. Thus, when it becomes necessary to refresh or invalidate a Web page, CachePortal.TM. 94 will send an invalidate or refresh request or message 242 to the appropriate location.

Detailed Description Text - DETX (14):

The difference between a refresh and invalidate request is that the refresh request will cause the cache 54 to request a new page from the Web server 58. The Web server 58 will then use its backend systems to retrieve the requested Web page and store it back into the cache 54. Alternatively, the new page can be prepared by the application server 70, the DBMS 74, or CachePortal.TM. 94. In contrast, by invalidating the Web page in cache 54, the Web page is simply tagged as invalid, expired, or to be deleted; no request is made for a new Web page. Instead, a new Web page will not be created until the next end user request for that invalid Web page. When the end user 52 requests the invalid Web page, the end user's request will generate a redirected Web page request to the Web server 58, which may take a long time to process. However, this long access time may not be a problem overall, if there are many subsequent users who request that same Web page. While the first requesting end user will endure a lengthy access time, the subsequent requests will get the benefit of the first end user's request because they can access the fresh content directly from cache 54. Therefore, overall, the invalidation method may be preferable for large numbers of end users 52.

Detailed Description Text - DETX (15):

The decision on whether to invalidate or refresh a Web page is a function of how often the Web page is accessed and the system resources, among other things. For example, if a Web page is accessed infrequently it may not make sense to refresh the Web page. Multiple refreshes of a Web page over a period of time are unnecessary, and a waste of resources, if no requests for that Web page occur during that period of time. In such a case, it may be beneficial to wait until a new request before refreshing the Web page. On the other hand, if frequent accesses to the Web page are expected, it may be beneficial to refresh the content immediately, because such a refresh will likely be needed in the near future.

Detailed Description Text - DETX (16):

Another reason why refreshing may not be the best alternative is that cache often uses a replacement strategy. That is, the cache can only store a limited amount of content, and if large amount of content is to be stored in cache, the cache will have to decide which stored object can be thrown out and replaced with new content. It may decide to throw out of cache the least frequently used content, or the content used least recently. However, by automatically refreshing a Web page in cache every time the data associated with that Web page changes, that Web page refreshed in cache will always look like it had been recently used. Thus, it may never get thrown out of cache even though it is infrequently accessed by end users.

Detailed Description Text - DETX (17):

As an alternative to a refresh message, a command can be sent from CachePortal.TM. 94 directly to the Web server 58, which will appear to the Web server 58 to be a request for the targeted Web page from an end user. This command will cause the Web server 58 to fetch a newly created dynamic Web page using its backend systems. This alternative may be more efficient because, if the cache 54 is remotely located, instructing the cache 54 to refresh the Web page may be undesirably slow. In addition, the cache 54 may be owned by a different entity, or it may be cache which has little processing capability other than storing content.

Detailed Description Text - DETX (20):

In alternative embodiments of the present invention, the content change monitoring component 104 may have access to the Web server log 68, and thus will be able to create the URL/Relevant operation mapping table 96 and determine which Web page to invalidate or maintain. If the content change monitoring component 104 additionally knows the location of the content stored in cache 54 or other Web server 58, the content change monitoring component 104 can even send the invalidate or refresh message to that location. Thus, it may be possible for either the content change monitoring component 104 or CachePortal.TM. 94 to send the invalidate or refresh messages. Furthermore, it should also be understood that content delivery service provider functionality and content change monitoring component 104 functionality (described in greater detail hereinafter) may be merged into the same server.

Detailed Description Text - DETX (26):



US006769025B1

(12) **United States Patent**
Alles et al.

(10) Patent No.: **US 6,769,025 B1**
(45) Date of Patent: ***Jul. 27, 2004**

(54) **FLEXIBLE EXTERNAL CONTROL OF
UNSOLICITED WEB PAGES SENT TO A
USER ACCESSING THE INTERNET**

6,397,246 B1 * 5/2002 Wolfe 709/217
6,430,739 B1 * 8/2002 Ballard 717/172
6,438,125 B1 * 8/2002 Brothers 709/219

(75) Inventors: **Anthony L. Alles**, Sunnyvale, CA (US);
Gil Tene, San Carlos, CA (US); **Shyam
Prasad Pillalamarri**, Palo Alto, CA
(US)

FOREIGN PATENT DOCUMENTS

WO WO 98/41913 * 9/1998

(73) Assignee: **Nortel Networks Limited**, St. Laurent
(CA)

OTHER PUBLICATIONS

HTML 4.0 Specification, Apr. 1998, World Wide Web
Consortium, section 7.*

(*) Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

* cited by examiner

Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

Primary Examiner—Zarni Maung

(74) *Attorney, Agent, or Firm—Blakely Sokoloff Taylor & Zafman LLP*

(57) **ABSTRACT**

An internet service node (ISN) for sending unsolicited web pages only to users agreeing to receive unsolicited web page. An internet service provider (ISP) may charge lower rates to users agreeing to receive unsolicited web pages. For example, an ISP may send an unsolicited web page in response to the first web page request for a user. In general, a user may agree to receive different types of unsolicited web pages under different conditions. An ISN may monitor the data flows related to the user to determine whether any condition is satisfied; and a web server may be used to generate and send the corresponding unsolicited web page.

(21) Appl. No.: **09/527,060**

(22) Filed: **Mar. 16, 2000**

(51) Int. Cl.⁷ **G06F 15/173**

(52) U.S. Cl. **709/225; 709/229; 709/230**

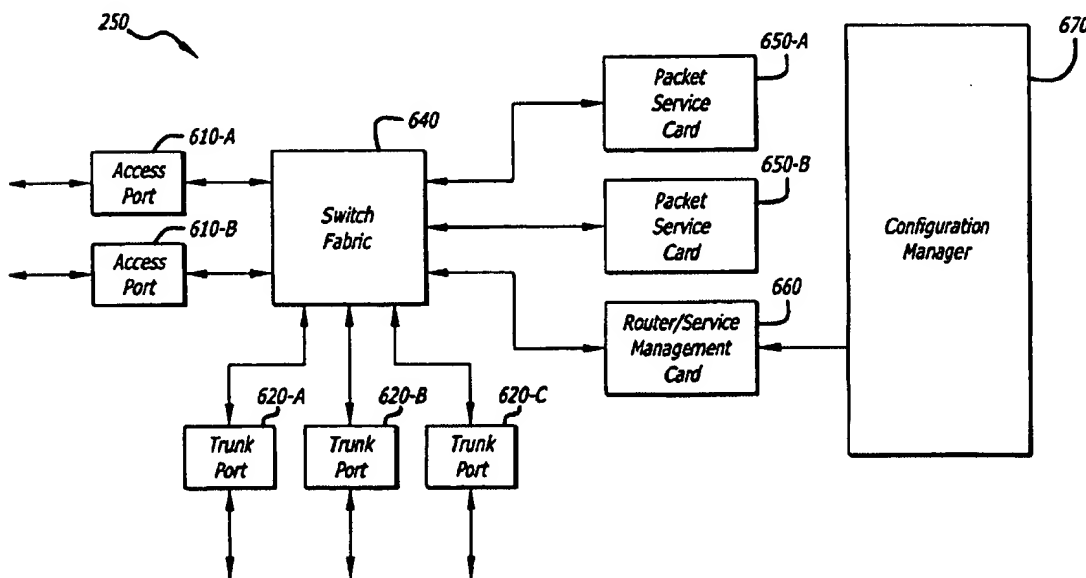
(58) Field of Search **709/217, 219,
709/225, 229, 203, 230**

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,219,042 B1 * 4/2001 Anderson et al. 345/716

17 Claims, 4 Drawing Sheets



US-PAT-NO: 6769025
DOCUMENT- US 6769025 B1
IDENTIFIER:
TITLE: Flexible external control of unsolicited web pages sent to a user accessing the internet

Brief Summary Text - BSTX (23):

The present invention provides a non-intrusive mechanism of sending web page with advertisements as the web page requested by the user will be automatically received by the user due to the URL specified for the unsolicited web page and the refresh period.

Detailed Description Text - DETX (46):

ISN 250 may determine whether the user has subscribed to the service of receiving additional web pages. If additional web pages can be sent, ISN 250 may forward the received web page access request to web server 260. Web server 260 may generate an unsolicited page with a pre-set automatic refresh period. As is well known in the relevant arts, conventional browsers (client software) request the web page with the same URL after a time duration specified by the automatic refresh period.

Claims Text - CLTX (7):

7. A method of serving web page access requests from a user, the method comprising: (a) configuring an access network indicating whether unsolicited web pages can be sent to said user; (b) receiving a web page access request from said user, said web page access request specifies a web page indicated by said user; and (c) sending an unsolicited web page in response to reception of said web page access request if the access network is configured so that unsolicited web pages can be sent to said user, said unsolicited web page is different from said web page specified by said web page access request and comprises a uniform resource locator (URL) of said web page indicated by said user, said unsolicited web page further comprises an automatic refresh period such that a user system used by said user automatically sends said web page access request again.